FIG. 1

50

```
<HTML>

<%@ taglib uri="/tlds/i2.tld" prefix="i2" %>        52a
<%@ page language="java" imports="java.util.*" %>   52b

<H1>Test Page</H1>     64

<P>Today is</P>     64
<jsp:useBean id="clock" class="jspCalendar" />    54
<UL>     64
<LI>Day: <%=clock.getDayOfMonth() %>  ] 56
<LI>Year: <%=clock.getYear() %>       ]
</UL>    64
<%--Check for AM or PM--%>
<%! int time=Calendar.getInstance().get(Calendar.AM_PM); %>    58

<%
if(time==Calendar.AM) {
%>
Good Morning
<%
}
else{                              } 60
%>
Good Afternoon
<%
}
%>

<P>Try this test button</P>    64
<i2:button onclick="javascript:alert('OUCH-stop that!')">Click Me</i2:button>
                                                                           62
<HTML>
```
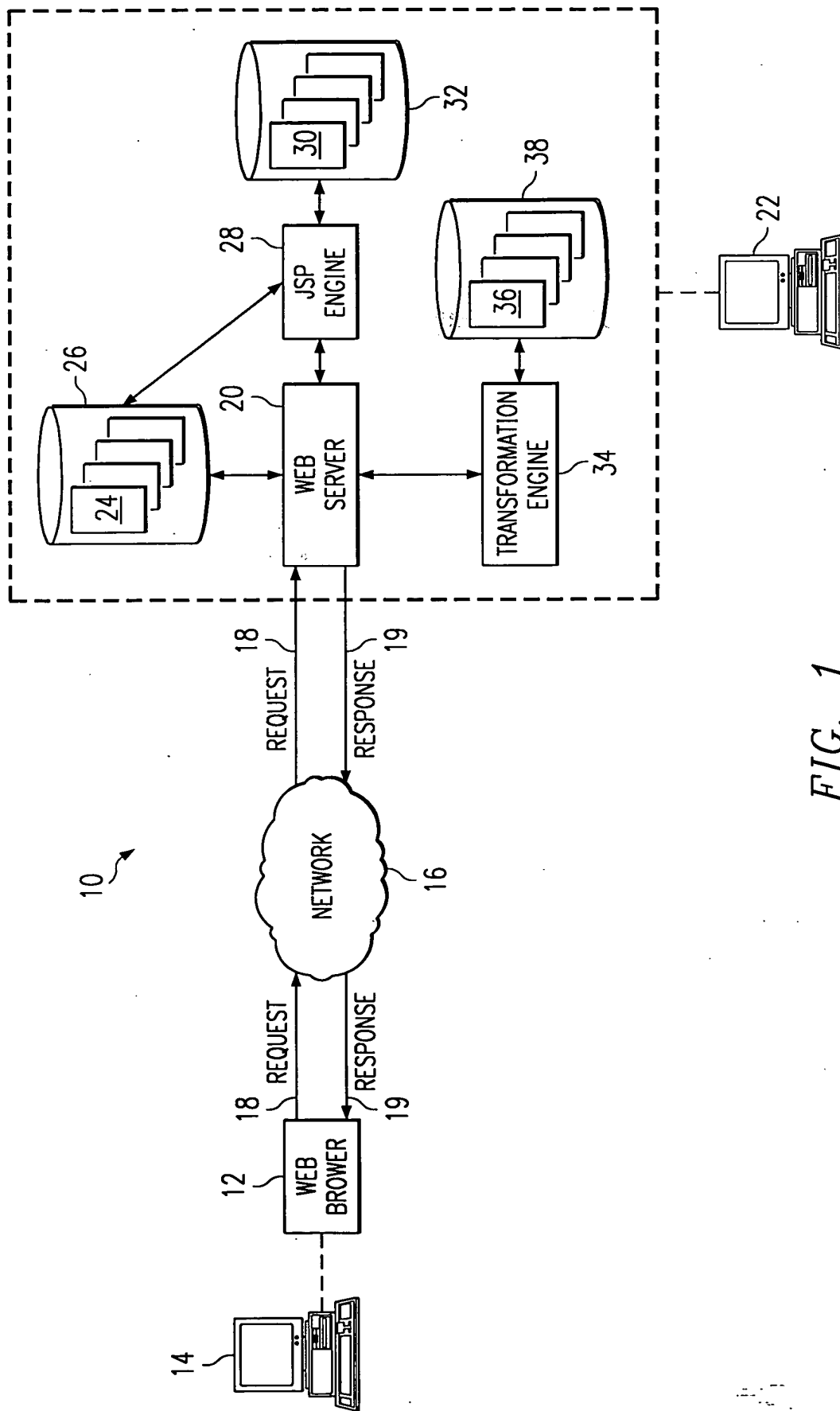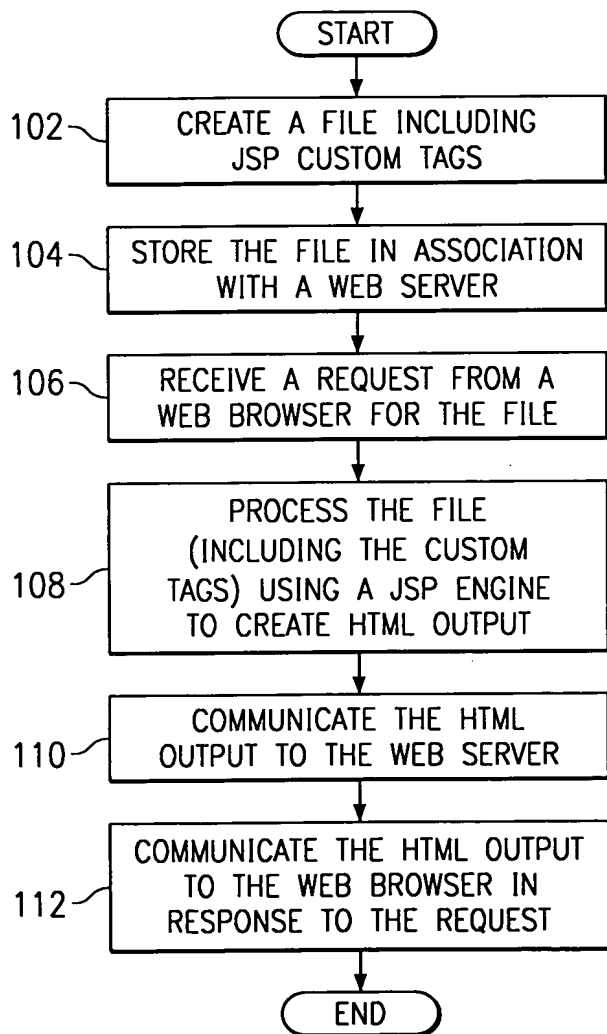
FIG. 2

FIG. 3

START

102 — CREATE A FILE INCLUDING JSP CUSTOM TAGS

104 — STORE THE FILE IN ASSOCIATION WITH A WEB SERVER

106 — RECEIVE A REQUEST FROM A WEB BROWSER FOR THE FILE

108 — PROCESS THE FILE (INCLUDING THE CUSTOM TAGS) USING A JSP ENGINE TO CREATE HTML OUTPUT

110 — COMMUNICATE THE HTML OUTPUT TO THE WEB SERVER

112 — COMMUNICATE THE HTML OUTPUT TO THE WEB BROWSER IN RESPONSE TO THE REQUEST

END

FIG. 7

START

162 — CREATE A FILE INCLUDING JSP CUSTOM TAGS

164 — STORE THE FILE IN ASSOCIATION WITH A WEB SERVER

166 — RECEIVE A REQUEST FROM A WEB BROWSER FOR THE FILE

168 — COMMUNICATE THE FILE, BASED ON A SPECIFIED CONTENT TYPE, TO A TRANSFORMATION ENGINE

170 — TRANSFORM THE CUSTOM TAGS TO CREATE HTML OUTPUT

172 — COMMUNICATE THE HTML OUTPUT TO THE WEB SERVER

174 — COMMUNICATE THE HTML OUTPUT TO THE WEB BROWSER IN RESPONSE TO THE REQUEST
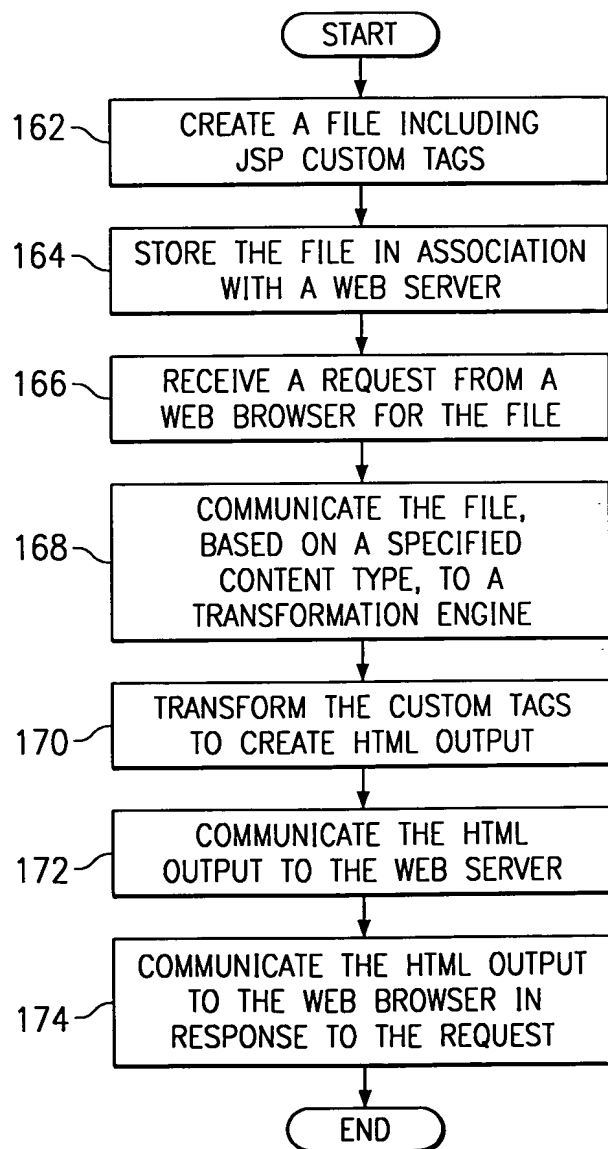
END

FIG. 4

36

```
<xsl:stylesheet
xmlns:xsl = "http://www.w3.org/1999.XSL/Transform"
xmlns:i2 = "http://www.i2.com"
exclude-result-prefixes = "i2"
result-ns = " ">                                              70

<xsl:output method = "html"/>

<xsl:template match="/">
 <xsl:apply-templates />                                      72a
<xsl:template>                            74

<xsl:template match = "i2:button">
 <TABLE width = "65" border = "1" cellspacing = "0" cellpadding = "2">
  <TR>
   <TD class = "buttonText" nowrap = "yes">      76
    <A class = "buttonText">
     <xsl:attribute name = "href">
      <xsl:value-of select = "@onclick"/>        78
     </xsl:attribute>
     <xsl:value-of select = "."/>
    </A>
   </TD>                                                      72b
  </TR>                              76
 </TABLE>
</xsl:template>

</xsl:stylesheet
```

FIG. 5

80

```
<TABLE width = "65" border = "1" cellspacing = "0" cellpadding = "2">
 <TR>
  <TD class = "buttonText" nowrap = "yes">
   <A class = "buttonText" href="javascript:alert('OUCH-stop that!')">Click Me</A>
  </TD>
 </TR>
</TABLE>
```
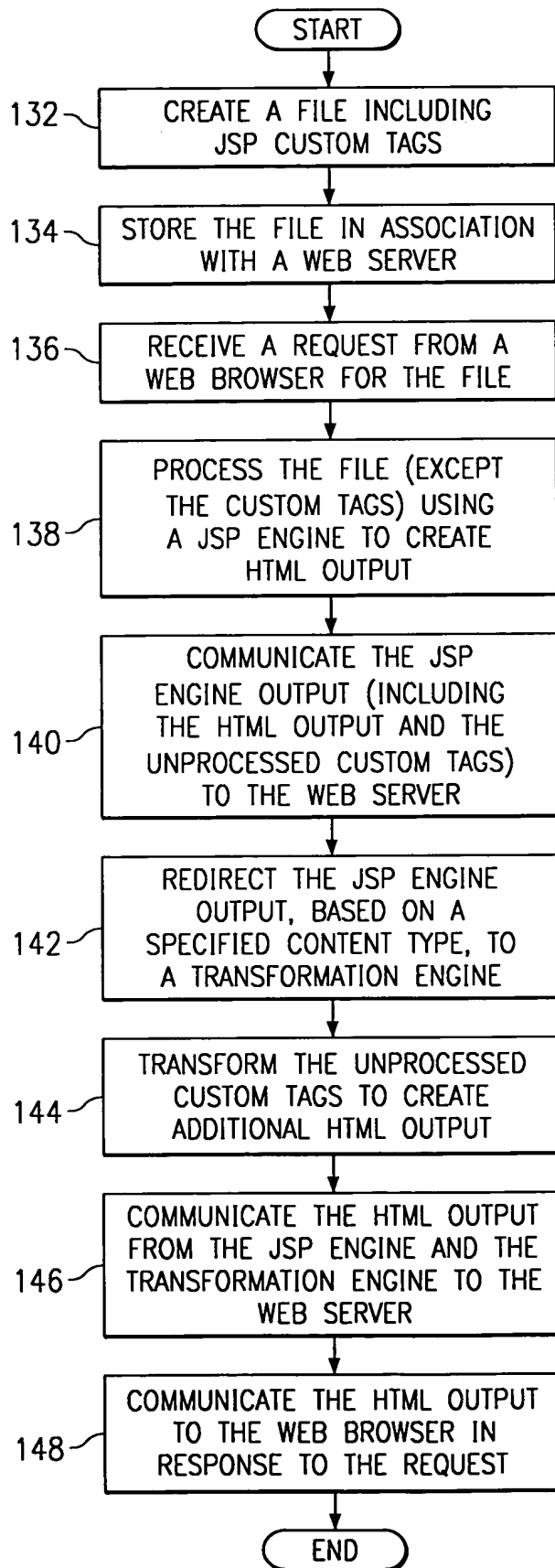
FIG. 6

START

132 — CREATE A FILE INCLUDING JSP CUSTOM TAGS

134 — STORE THE FILE IN ASSOCIATION WITH A WEB SERVER

136 — RECEIVE A REQUEST FROM A WEB BROWSER FOR THE FILE

138 — PROCESS THE FILE (EXCEPT THE CUSTOM TAGS) USING A JSP ENGINE TO CREATE HTML OUTPUT

140 — COMMUNICATE THE JSP ENGINE OUTPUT (INCLUDING THE HTML OUTPUT AND THE UNPROCESSED CUSTOM TAGS) TO THE WEB SERVER

142 — REDIRECT THE JSP ENGINE OUTPUT, BASED ON A SPECIFIED CONTENT TYPE, TO A TRANSFORMATION ENGINE

144 — TRANSFORM THE UNPROCESSED CUSTOM TAGS TO CREATE ADDITIONAL HTML OUTPUT

146 — COMMUNICATE THE HTML OUTPUT FROM THE JSP ENGINE AND THE TRANSFORMATION ENGINE TO THE WEB SERVER

148 — COMMUNICATE THE HTML OUTPUT TO THE WEB BROWSER IN RESPONSE TO THE REQUEST

END